

CENG 351

Data Management and File Structures

Fall 2020

Programming Assignment 1

Due date: 22.12.2020, Tuesday, 23:59

1 Introduction

In this assignment, you are asked to create a system by designing queries and implementing pre-defined functions to operate on a database for a video streaming service called CengTube. You will have certain tasks and a well-defined interface.

What you will do is to implement the provided interface to accomplish the given tasks. All necessary data files, classes and the interface you will implement are provided to you in the source files. Do not confuse interface with graphical user interface (GUI). Interface is an abstract type in Java used to specify a behavior that classes must implement. (If you remember from the Programming Language Concepts course, a Java interface corresponds to a C++ class that all of its methods are pure virtual.)

The first thing you should do is implementing functions which create the necessary tables corresponding to the schema given in Section 3. Then, you should design queries to accomplish the given tasks. Lastly, you should implement the interface using the queries you have designed as they give the desired results when defined parameters are given. You will not implement the Evaluation class. It will be implemented by us to manipulate the database through the predefined interface and evaluate your implementations. Your task is to build up class(es) which implement(s) the provided interface.

2 Objectives

This assignment aims to help you get familiar with

- Connecting and querying to MySQL Server using Java Database Connectivity (JDBC)

3 Schema

You will use (strictly) the schema given below in the scope of this assignment.

User(userID:int, userName:varchar(30), email:varchar(30), password:varchar(30), status:varchar(15))

Video(videoID:int, userID:int, videoTitle:varchar(60), likeCount:int, dislikeCount:int, datePublished:date)

Comment(commentID:int, userID:int, videoID:int, commentText:varchar(1000), dateCommented:date)

Watch(userID:int, videoID:int, dateWatched: date)

Your task is to generate a class named CengTubeDB (it should belong to package ceng.ceng351.cengtubedb) which implements ICengTubeDB interface. You can create any additional classes if necessary. CengTubeDB class should be able to accomplish the following tasks:

- Creating the database tables
- Inserting data into tables
- List videos which have higher likeCount than dislikeCount
- List videos commented by the given userID
- Find the oldest published video for a given user which doesn't contain 'VLOG' in its title
- List the trending top three videos for a given time interval (detailed explanation is given below)
- List users and number of videos watched only by her/him

- List users who have watched and commented all of their own videos
- List the users that has at least one watch but no comments
- Update the users' status to "verified" if the view count of all their videos in total is more than the given number
- Given a video id and a string, update the title of the video with the given id to the new string
- Given video title, delete that video from the database
- Dropping the database tables

Tasks are explained in more detail below. For each task, there is a corresponding method in ICengTubeDB interface. You need to implement them in CengTubeDB class. Necessary data files (for populating the tables) to accomplish these tasks are provided.

In **data** folder there are 4 txt files that correspond to each table. You will use these tables when you are inserting data. Data files, interfaces and classes for fulfilling these tasks will be provided as source files. You can assume all information will be complete and consistent, i.e. all necessary data will be inserted before executing a query. You can find detailed description about the usage of the functions in provided source files. Your results should not include any repetition. Therefore, please do not forget to use DISTINCT keyword when appropriate in your queries.

Make sure you are using Java(version13) before starting.

3.1 Creating the database tables (10 pts)

You will create all the tables according to the schema described above.

You can assume that tables will be created before executing any other database operation.

Do not forget to define **foreign keys** while you are creating tables.

Also, you need to consider that whenever a video is deleted, comments and watches related to it should also be deleted. Moreover, whenever a user deleted, videos and watches related to it should also be deleted and userID in comments related to it should be set to NULL.

Output: the number of tables that are created successfully.

3.2 Inserting data into tables (5 pts)

You will insert data into appropriate tables.

Output: the number of rows inserted successfully.

3.3 List videos which have higher likeCount than dislikeCount (5 pts)

Output: videoTitle, likeCount, dislikeCount

Order the results by videoTitle in ascending order.

3.4 List the videos commented by the given userID (5 pts)

Input: userID

Output: videoTitle, userName, commentText

Order the results by videoTitle in ascending order.

3.5 Find the oldest published video for a given userID which doesn't contain "VLOG" in its title (5 pts)

Input: userID

Output: videoTitle, userName and datePublished.

Order the results by videoTitle in ascending order.

3.6 List the trending top three videos for a given time interval (10 pts)

A trending video is defined to be the most viewed video in the given interval (i.e., video that is viewed the highest number of times among all).

You should include dateStart and dateEnd in the result, it is a CLOSED interval.

Input: startDate, endDate

Output: videoTitle, userName, number of times that the video watched.

Order the results by number of times that the video watched in **descending** order.

3.7 List users and number of videos watched only by her/him (10 pts)

Output: userID, userName, number of videos watched by the user(only count the ones that match the criteria)
Order the results by userID in ascending order.

3.8 List users who have watched and commented all of their own videos (10 pts)

Output: userID, userName, email
Order the results by userID in ascending order

3.9 List users that has at least one watch but no comments (10 pts)

Output: userID, userName, email
Order the results by userID in ascending order

3.10 Update the users' status to "verified" if the view count of all their videos in total is more than the given number (10 pts)

View count simply means how many times that a video has watched.
Input: givenViewCount
Output: number of rows affected.

3.11 Given a video id and a string, update the title of the video with the given id to the new string (10 pts)

Input: videoID, newTitle
Output: number of rows affected.

3.12 Given video title, delete that video(s) from the database(5 pts)

Note that related comments of the video(s) should be deleted as well. You should handle that by the design of your schema.
Input: videoTitle
Output: number of rows in the video table after delete operation.

3.13 Dropping the database tables (5 pts)

You will drop all the tables (if they exist).
Output: number of tables that are dropped successfully.

4 Regulations

1. Programming Language: Java(Version13).
2. Database: An account on the MySQL server on a remote machine will be created for each of you and an e-mail including credentials and connection configuration will be sent to your metumail. You must use JDBC driver to connect to the database. Your final submission must connect to the MySQL server on the remote machine. So, make sure that the connection information is correct before submitting your homework.
3. Attachments: Necessary source files and JDBC driver is provided.
4. Input: All strings will be case-sensitive and they will not include any non-English characters. Note that they may include punctuation.
5. Cheating: We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations.
6. Evaluation: It is GUARANTEED that input files are correctly formatted and sample data will be given to you. There will be no surprises about the data, similar (and larger) data will be used while evaluating homeworks. Your program will be evaluated automatically using "black-box" technique so make sure to obey the specifications. Please, be noticed that you have to accomplish tasks only within your sql queries not with any other Java programming facilities.

5 Submission

Submission will be done via ODTUClass. Create a compressed file named `ceng.tar.gz` that contains `CengTubeDB.java` and all other classes created by yourself. You will not submit interface and class files provided by me. So, be sure you do not modify them during implementation. The compressed file should contain a directory tree same as the package tree. That is, you should compress the directory named 'ceng' which contains a directory named 'ceng351' which contains a directory named 'cengtubedb' which contains your source files.

```
ceng
├── ceng351
│   └── cengtubedb
│       ├── CengTubeDB.java
│       ├── AnotherClassIfYouNeed1.java
│       ├── AnotherClassIfYouNeed2.java
│       ├── ..
│       ├── ...
│       └── AnotherClassIfYouNeedN.java
```